SOURCECODE

module alu (

input [7:0] a,b,

input [3:0] Alu\_Sel,

output [7:0] Alu\_Out,

output CarryOut

);

reg [7:0] Alu\_Result;

wire [8:0] tmp;

assign Alu\_Out = Alu\_Result;

assign tmp = {1'b0,a}+{1'b0,b};

assign CarryOut = tmp [8];

always @ (\*)

begin

case (Alu\_Sel)

4'b0000:

Alu\_Result = a+b;

4'b0001:

Alu\_Result = a-b;

4'b0010:

Alu\_Result = a\*b;

4'b0011:

Alu\_Result = a/b;

4'b0100:

Alu\_Result = a&b;

4'b0101:

Alu\_Result = a|b;

4'b0110:

Alu\_Result = a^b;

4'b0111:

Alu\_Result = ~(a|b);

4'b1000:

Alu\_Result = ~(a&b);

4'b1001:

Alu\_Result = ~(a^b);

4'b1010:

Alu\_Result = (a>b);

default : Alu\_Result = a+b;

endcase

end

endmodule

TEST BENCH

`timescale 1ns/1ps

module alu\_tb;

reg [7:0] a,b;

reg [3:0] Alu\_Sel;

wire [7:0] Alu\_Out;

wire CarryOut;

integer i;

alu test\_unit (a,b,Alu\_Sel,Alu\_Out,CarryOut);

initial begin

$dumpfile ("alu.vcd");

$dumpvars (1,alu\_tb);

a = 8'h05;

b = 8'h02;

Alu\_Sel = 4'h0;

for (i=0;i<4;i=i+1)

begin

Alu\_Sel = Alu\_Sel+8'h01;

#10;

end;

a = 8'hFF;

b = 8'h00;

end

endmodule



